**Web漏洞攻击**

**跨站请求伪造攻击实践**

**第一步：访问DVWA并选择漏洞**

**1. 打开浏览器，输入DVWA的URL。**

**2. 使用默认的用户名`admin`和密码`password`登录。**

**3. 在DVWA的安全设置中，将安全级别调整为`medium`。**

**4. 选择`CSRF`（Cross Site Request Forgery）漏洞进行实践。**

**目的：配置实验环境，确保可以进行CSRF攻击。**

**第二步：分析利用漏洞需要的参数**

**1. 观察页面中的表单，通常包含用户操作需要提交的数据。**

**2. 使用网络抓包工具（如Burp Suite）监控提交表单时的HTTP请求，了解哪些参数是必须的。**

**目的：理解正常用户操作时数据的提交方式，为构造攻击请求做准备。**

**第三步：构造CSRF攻击请求**

**1. 根据抓包得到的参数，构造一个链接，当受害者点击这个链接时，会自动提交一个带有攻击者参数的请求。**

**2. 确保这个链接能够触发敏感操作，比如修改用户信息、转账等。**

**目的：制作一个看似无害但实际含有攻击代码的链接，诱导受害者点击，达到攻击目的。**

**第四步：诱导受害者点击链接**

**- 通过电子邮件、社交媒体消息等方式发送给受害者。**

**目的：让受害者在不知情的情况下执行了攻击者构造的请求。**

**第五步：验证攻击是否成功**

**- 检查受害者账户是否发生了预期的更改。**

**目的：确认CSRF攻击是否成功执行。**

**文件上传漏洞攻击实践**

**第一步：访问DVWA并选择漏洞**

**1. 同上述CSRF攻击的第一步。**

**2. 选择`File Upload`漏洞进行实践。**

**目的：配置实验环境，确保可以进行文件上传攻击。**

**第二步：尝试上传恶意文件**

**1. 浏览到文件上传页面，选择一个恶意文件（如PHP后门）。**

**2. 上传该文件，观察是否能成功。**

**目的：测试应用是否允许上传恶意文件，这是文件上传攻击的关键一步。**

**第三步：分析上传文件后的响应**

**- 如果上传成功，记下上传后文件的URL。**

**- 如果上传失败，观察失败的原因，可能是文件类型、大小限制等。**

**目的：了解应用对上传文件的限制，为下一步攻击做准备。**

**第四步：访问上传的恶意文件**

**- 如果上传的是PHP后门，尝试直接访问其URL，看是否能执行代码。**

**目的：验证上传的文件是否能够被服务器执行，实现远程代码执行的目的。**

**第五步：如果需要，尝试绕过限制**

**- 如果发现有文件类型或大小的检测，尝试使用工具或技巧绕过这些限制。目的：提高攻击成功率，确保能够上传并执行恶意代码。**